

[Date of requesting appeal against examiner's decision of rejection]	22.05.1996
[Date of extinction of right]	25.08.2004

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-84848

(43)公開日 平成7年(1995)3月31日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/00	5 3 1 J	8944-5B		
1/00	3 7 0 D			
9/06	4 1 0 B	9367-5B		
9/445				
		9367-5B	G 0 6 F 9/ 06	4 2 0 L
審査請求 有 請求項の数19 OL (全 24 頁) 最終頁に続く				

(21)出願番号 特願平5-184186

(22)出願日 平成5年(1993)7月26日

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーションINTERNATIONAL BUSIN
ESS MASCHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72)発明者 下遠野 享

神奈川県大和市下鶴間1623番地14 日本ア

イ・ビー・エム株式会社 大和事業所内

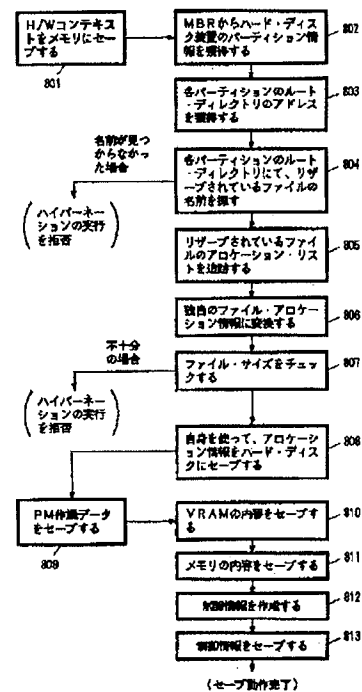
(74)代理人 弁理士 合田 潔 (外3名)

(54)【発明の名称】 情報処理システム

(57)【要約】 (修正有)

【目的】ハイパーネーション・ファイルシステムで、ハイパーネーション及びブウェーク・アップの一連の動作を高速に行う。

【構成】ハイパーネーション・モードに入るときは、外部記憶装置上の、OSファイル・システムが管理する情報からハイパーネーション・ファイルのアロケーション情報を獲得し、バッファに入力する(ステップ805、806)。その情報を用いてメイン・メモリ及びVRAMの内容並びにアロケーション情報をハイパーネーション・ファイルにセーブする(ステップ808、810、811)。また、外部記憶装置上のアロケーション情報がセーブされたアドレス情報を所定の制御情報領域にストアする(ステップ813)。ウェーク・アップするときには、ハイパーネーション・ファイル中に既にあるアロケーション情報を使ってメイン・メモリ及びVRAMの内容をリストアする。



【特許請求の範囲】

【請求項 1】CPU、揮発性のメイン・メモリ及び不揮発性の外部記憶装置を備え、所定の状態が発生したときにタスクを中断し、上記メイン・メモリの内容を、上記外部記憶装置に存在し、OSファイル・システムによって管理されているハイパーネーション・ファイルにセーブして上記メモリへの給電を停止するハイパーネーション機能をサポートする情報処理システムであって、

(a) ハイパーネーション・モードに入るときに、上記外部記憶装置上の、上記OSファイル・システムが管理するファイル・アロケーション情報をストアしている領域にアクセスして、上記ハイパーネーション・ファイルのアロケーション情報を獲得し、上記システム中のバッファに入力する手段と、(b) 上記バッファ中の上記アロケーション情報をそれ自身を用いて上記ハイパーネーション・ファイルにセーブする手段と、(c) 上記バッファ中の上記アロケーション情報を用いて、上記メイン・メモリの内容を、上記ハイパーネーション・ファイルにセーブする手段と、(d) 上記外部記憶装置上の上記アロケーション情報がセーブされた場所のアドレス情報を、上記外部記憶装置上の所定の領域にストアする手段を具備する、情報処理システム。

【請求項 2】上記手段 (a) は、上記OSファイル・システムが管理するハイパーネーション・ファイルのアロケーション情報を、上記外部記憶装置上の当該ファイルを構成する一続きのセクタ・ブロックごとにそのスタート・アドレスと長さを示す情報に変換し、変換後の情報を上記バッファに入力することを特徴とする、請求項 1 記載の情報処理システム。

【請求項 3】CPU、揮発性のメイン・メモリ及びビデオ・メモリ並びに不揮発性の外部記憶装置を備え、所定の状態が発生したときに、上記メイン・メモリ及び上記ビデオ・メモリの内容を、上記外部記憶装置に存在し、OSファイル・システムによって管理されているハイパーネーション・ファイルにセーブしてシステム全体をパワーオフするハイパーネーション機能をサポートする情報処理システムであって、(a) ハイパーネーション・モードに入るときに、上記外部記憶装置上の、上記OSファイル・システムが管理するファイル・アロケーション情報をストアしている領域にアクセスして、上記ハイパーネーション・ファイルのアロケーション情報を獲得し、上記システム中のバッファに入力する手段と、

(b) 上記バッファ中の上記アロケーション情報をそれ自身を用いて上記ハイパーネーション・ファイルにセーブする手段と、(c) 上記バッファ中の上記アロケーション情報を用いて、上記メイン・メモリ及び上記ビデオ・メモリの内容を、上記ハイパーネーション・ファイルにセーブする手段と、(d) 上記外部記憶装置上の上記アロケーション情報がセーブされた場所のアドレス情報を、上記外部記憶装置上の所定の制御情報領域にストア

する手段を具備する、情報処理システム。

【請求項 4】上記手段 (a) は、上記OSファイル・システムが管理するハイパーネーション・ファイルのアロケーション情報を、上記外部記憶装置上の当該ファイルを構成する一続きのセクタ・ブロックごとにそのスタート・アドレスと長さを示す情報に変換し、変換後の情報を上記バッファに入力することを特徴とする、請求項 3 記載の情報処理システム。

【請求項 5】上記ハイパーネーション・ファイル中で、上記メイン・メモリの内容と上記ビデオ・メモリの内容は区別して管理されており、上記制御情報領域には、上記外部記憶装置上の上記メイン・メモリの内容がセーブされた場所と上記ビデオ・メモリの内容がセーブされた場所を示すアドレス情報がストアされることを特徴とする、請求項 3 記載の情報処理システム。

【請求項 6】ハードウェア・コンテキスト情報を含む作業データを上記ハイパーネーション・ファイルにセーブする手段を具備し、

上記制御情報領域には、上記外部記憶装置上の上記作業データがセーブされた場所を示すアドレス情報がストアされることを特徴とする、請求項 3 記載の情報処理システム。

【請求項 7】システム・コンフィギュレーション情報をストアするCMOS手段と、
上記システム・コンフィギュレーション情報を上記CMOS手段から上記制御情報領域にストアする手段とを具備することを特徴とする、請求項 3 記載の情報処理システム。

【請求項 8】ハイパーネーション・モードに入るための一連のセーブ動作を経てパワーオフされたことを示すハイパーネーション・シグニチャを上記制御情報領域にストアする手段を具備することを特徴とする、請求項 3 記載の情報処理システム。

【請求項 9】システムのパワーオンまたはリセット時に、上記ハイパーネーション・シグニチャの有無に基づいて、ノーマル・ブートまたは上記外部記憶装置からのリストアを伴うブートの何れかを選択する手段を具備することを特徴とする、請求項 8 記載の情報処理システム。

【請求項 10】上記ハイパーネーション・ファイルのサイズをチェックして、サイズが不十分である場合には、ハイパーネーションの実行を拒否する手段を具備することを特徴とする、請求項 3 記載の情報処理システム。

【請求項 11】CPU、揮発性のメイン・メモリ及び不揮発性の外部記憶装置を備え、ハイパーネーション・モード中にパワーがオンになったことに応答して、上記メイン・メモリの内容を上記外部記憶装置上のOS管理下のハイパーネーション・ファイルからリストアし、中断されていたタスクを再開するウェーク・アップ機能をサポートする情報処理システムであって、

上記ハイパーネーション・ファイルは、当該ファイルのアロケーション情報を含んでおり、

上記ハイパーネーション・ファイル中の上記アロケーション情報が存在する場所のアドレス情報を含む制御情報が、上記外部記憶装置上の所定の制御情報領域にストアされており、

上記情報処理システムは、(a) 上記制御情報領域にアクセスして上記アドレス情報を獲得し、上記アロケーション情報を上記システム中のバッファに入力する手段と、(b) 上記バッファ中の上記アロケーション情報を用いて、上記メイン・メモリの内容を上記ハイパーネーション・ファイルからリストアする手段を具備することを特徴とする、情報処理システム。

【請求項 12】 上記ハイパーネーション・ファイル中の当該ファイルのアロケーション情報は、上記外部記憶装置上での当該ファイルを構成する一続きのセクタ・ブロックごとにそのスタート・アドレスと長さを示す情報であることを特徴とする、請求項 11 記載の情報処理システム。

【請求項 13】 ウェーク・アップのための一連の処理の最後に、上記制御情報を無効化する手段を具備することを特徴とする請求項 11 記載の情報処理システム。

【請求項 14】 CPU、揮発性のメイン・メモリ及びビデオ・メモリ並びに不揮発性の外部記憶装置を備え、ハイパーネーション・モード中にパワーがオンになったことに応答して、上記メイン・メモリ及び上記ビデオ・メモリの内容を上記外部記憶装置上の OS 管理下のハイパーネーション・ファイルからリストアし、中断されていたタスクを再開するウェーク・アップ機能をサポートする情報処理システムであって、

上記ハイパーネーション・ファイルは、当該ファイルのアロケーション情報を含んでおり、

上記ハイパーネーション・ファイル中の上記アロケーション情報が存在する場所のアドレス情報を含む制御情報が、上記外部記憶装置上の所定の制御情報領域にストアされており、

上記情報処理システムは、(a) 上記制御情報領域にアクセスして上記アドレス情報を獲得し、上記アロケーション情報を上記システム中のバッファに入力する手段と、(b) 上記バッファ中の上記アロケーション情報を用いて、上記メイン・メモリ及び上記ビデオ・メモリの内容を上記ハイパーネーション・ファイルからリストアする手段を具備することを特徴とする、情報処理システム。

【請求項 15】 上記ハイパーネーション・ファイル中の当該ファイルのアロケーション情報は、上記外部記憶装置上での当該ファイルを構成する一続きのセクタ・ブロックごとにそのスタート・アドレスと長さを示す情報であることを特徴とする、請求項 14 記載の情報処理システム。

【請求項 16】 上記ハイパーネーション・ファイル中で、上記メイン・メモリの内容と上記ビデオ・メモリの内容は区別して管理されており、上記制御情報領域には、上記メイン・メモリの内容がセーブされている場所と上記ビデオ・メモリの内容がセーブされている場所を示すアドレス情報がストアされており、

上記手段 (b) は、上記メイン・メモリの内容についてのアドレス情報及び上記ビデオ・メモリの内容についてのアドレス情報を用いることを特徴とする、請求項 14 記載の情報処理システム。

【請求項 17】 上記ハイパーネーション・ファイル中に、ハードウェア・コンテキスト情報を含む作業データがセーブされ、上記制御情報領域には、上記作業データがセーブされている場所を示すアドレス情報がストアされており、

上記システムは、上記アロケーション情報及び上記作業データについてのアドレス情報を用いて、上記作業データを上記ハイパーネーション・ファイルからリストアする手段を具備することを特徴とする、請求項 14 記載の情報処理システム。

【請求項 18】 上記制御情報領域には、ハイパーネーション実行時のシステム・コンフィギュレーション情報がストアされており、

上記システムは、現在のシステム・コンフィギュレーションとハイパーネーション実行時のシステム・コンフィギュレーションを比較し、不一致のときはウェーク・アップ・シーケンスをストップすることを特徴とする、請求項 14 記載の情報処理システム。

【請求項 19】 ウェーク・アップのための一連の処理の最後に、上記制御情報を無効化する手段を具備することを特徴とする請求項 14 記載の情報処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、低消費電力で動作する情報処理システムに関し、特にノートブック・コンピュータのような携帯型の情報処理装置に関するものである。

【0002】

【従来の技術】 最近の技術革新によって、小型・軽量のポータブル・コンピュータが主流となってきた。このようなポータブル・コンピュータを室外で使用する場合、通常はバッテリーで電力を供給している。しかし、ポータブル・コンピュータに搭載するバッテリーは小型なものに限られるため、1回の充電でパソコンが動作可能な時間は短い。このため、多くのポータブル・コンピュータは、消費電力を減らすために種々の工夫を施している。

【0003】 サスペンド／レジューム機能は、そのような工夫の一つである。それによれば、一定時間 I/O デバイスのアクティビティが検出されない等の状態が発

生すると、コンピュータはサスペンド・モードに入る。サスペンド・モードの間、すべてのタスクは停止され、後で再開するのに必要なデータがメイン・メモリにセーブされる。サスペンド・モードの間、メイン・メモリ及びビデオ・メモリ（VRAM）への給電は行われるけれども、CPU等への給電は停止される。しかしながら、ポータブル・コンピュータにおいてサスペンド／レゾーム機能がサポートされているだけでは、サスペンド・モードが長く継続したときに、バッテリーの電力が消費され、その結果、メモリやVRAMの内容が失われるという問題点があった。

【0004】そこで、コンパック社のLTE Lite / 25のように、ハイパーネーション機能をサポートしたポータブル・コンピュータが提案され、販売されるようになった。（コンパック、LTEは、コンパック・コンピュータ・コーポレーションの商標である。）ハイパーネーションがイネーブルされると、ロー・バッテリー状態になったり、あるいはサスペンド・モードが一定時間継続するなどの状態が発生したときに、システムは、後でタスクを再開するのに必要なすべてのデータをハード・ディスクにセーブした後、ハイパーネーション・モードに入る。ハイパーネーション・モードでは、メモリ及びVRAMを含むシステム全体のパワーがオフされる。後でユーザがシステムのパワーをオンにすると、ハード・ディスクにストアされていたデータはメモリやVRAMにリストアされ、中断していたタスクが自動的に再開する。パワーオンに伴う一連の動作はウェイク・アップとも呼ばれる。

【0005】

【発明が解決しようとする課題】ハイパーネーションをサポートするためには、予めハード・ディスク装置にメイン・メモリやVRAMの内容をセーブするためのスペースを確保する必要がある。現在知られているシステムのあるものは、ハード・ディスク上にハイパーネーション専用の領域が確保されており、ユーザがその設定を変更することができない。しかしながら、そのようなシステムでは、セーブ可能な情報量が制限されるため、メイン・メモリを増設することが不可能である。

【0006】そこで、メイン・メモリなどの内容をセーブする宛先を、OSファイル・システムによって管理されるハイパーネーション・ファイルとするシステムが知られている。これによって、メイン・メモリが増設され、したがって外部記憶装置上に確保すべきスペースのサイズが増えた場合でも、スペースを確保し直すことが簡単にできる。

【0007】そのようなシステムでは、メモリやVRAMとハード・ディスク装置との間でデータ転送を行うために、OSによって管理されるハイパーネーション・ファイルのアロケーション・リストにアクセスすることが必須となる。一般に、ハイパーネーション・ファイルの

サイズは巨大であるため、アロケーション・リストも長大となる。ハイパーネーション／ウェイク・アップのための一連の動作を高速に行うためには、ハイパーネーション・ファイルのアロケーション情報を獲得するためにハード・ディスク装置にアクセスする頻度を可能な限り減らさなければならない。

【0008】本発明は、以上に鑑みなされたものであり、ハイパーネーション及びウェイク・アップのための一連の動作を高速で行うことを目的とする。

【0009】

【課題を解決するための手段】上記目的を達成するために、本発明の第1の側面は、CPU、揮発性のメイン・メモリ及びビデオ・メモリ並びに不揮発性の外部記憶装置を備え、所定の状態が発生したときに、上記メイン・メモリ及びビデオ・メモリの内容を、上記外部記憶装置に存在し、OSファイル・システムによって管理されているハイパーネーション・ファイルにセーブしてシステム全体をパワーオフするハイパーネーション機能をサポートする情報処理システムであって、（a）ハイパーネーション・モードに入るときに、上記外部記憶装置上の、上記OSファイル・システムが管理するファイル・アロケーション情報をストアしている領域にアクセスして、上記ハイパーネーション・ファイルのアロケーション情報を獲得し、上記システム中のバッファに入力する手段と、（b）上記バッファ中の上記アロケーション情報をそれ自身を用いて上記ハイパーネーション・ファイルにセーブする手段と、（c）上記バッファ中の上記アロケーション情報を用いて、上記メイン・メモリ及び上記ビデオ・メモリの内容を、上記ハイパーネーション・ファイルにセーブする手段と、（d）上記外部記憶装置上の上記アロケーション情報がセーブされた場所のアドレス情報を、上記外部記憶装置上の所定の制御情報領域にストアする手段を具備することを特徴とする。

【0010】また、本発明の第2の側面は、CPU、揮発性のメイン・メモリ及びビデオ・メモリ並びに不揮発性の外部記憶装置を備え、ハイパーネーション・モード中にパワーがオンになったことに応答して、上記メイン・メモリ及び上記ビデオ・メモリの内容を上記外部記憶装置上のOS管理下のハイパーネーション・ファイルからリストアし、中断されていたタスクを再開するウェイク・アップ機能をサポートする情報処理システムであって、上記ハイパーネーション・ファイルは、当該ファイルのアロケーション情報を含んでおり、上記ハイパーネーション・ファイル中の上記アロケーション情報が存在する場所のアドレス情報を含む制御情報が、上記外部記憶装置上の所定の制御情報領域にストアされており、上記情報処理システムは、（a）上記制御情報領域にアクセスして上記アドレス情報を獲得し、上記アロケーション情報を上記システム中のバッファに入力する手段と、（b）上記バッファ中の上記アロケーション情報を用い

て、上記メイン・メモリ及び上記ビデオ・メモリの内容を上記ハイパーネーション・ファイルからリストアする手段と、(c)上記制御情報を無効化する手段を具備することを特徴とする。

【0011】

【実施例】

A. システム全体の構成

図1は、簡略化された形で、本発明を具体化したノート・ブック型コンピュータ（以下では単にシステムと呼ぶ）の主要なハードウェア構成要素を示している。10はメインCPUであり、実施例ではメモリ・コントローラが内蔵されているインテル社の80486SLを用いている。CPU10は、メモリ・バス11を介してメイン・メモリ12及びPMメモリ13と連絡している。メイン・メモリ12には、BIOS、ドライバ、OS及びアプリケーションがロードされる。一方、PMメモリ13には、ハイパーネーションを含むパワー・マネジメントを行うPMコード（PMC）及びその作業データがストアされる。PMメモリはさらにPMCがストアされる領域と作業データがストアされる領域に分かれる。PMCは、POR（パワーオン／リセット）時にROMからロードされる。

【0012】別個のチップをPMメモリとメイン・メモリに割り当てること、例えばPMメモリにはSRAMチップを使い、メイン・メモリにはDRAMチップを使うことも可能であるが、実施例では、80486SLのアーキテクチャを用いて、1つのDRAMチップのそれぞれ特定の領域をメイン・メモリ12とPMメモリ13とに割り当てている。

【0013】PMCは、メイン・メモリ12とPMメモリ13の両方にアクセスすることができる。これに対し、OSやドライバはPMメモリ13にアクセスできない。メインCPU10がメイン・メモリ12とPMメモリ13のどちらとコミュニケーションするかは、メモリ・コントローラによって切り換えられる。

【0014】CPU10は、アドレス／データ・バス14を介して、トラップ・ロジック16、DMAC（ダイレクト・メモリ・アクセス・コントローラ）18、PIC（プログラム可能割り込みコントローラ）20、PIT（プログラム可能インターバル・タイマー）22、シリアル・ポート24、パラレル・ポート26、RTC（リアル・タイム・クロック）28、CMOS30、ROM32と結合している。

【0015】トラップ・ロジック16の出力端子とCPU10の特定のピンとはシステム割り込み線52によって接続されている。トラップ・ロジック16はバス14を常時モニタしており、その内蔵レジスタにセットされたアドレスへのアクセスを検出したときに、システム割り込み線52をアクティブにする。また、外部入力端子に入力される信号50がアクティブになったときにも、シ

テム割り込み線52をアクティブにする。

【0016】実施例では、インテル社のI/Oチップ・セット82360SLを用いている。これは、トラップ・ロジック16、DMAC18、PIC20、PIT22、シリアル・ポート24、RTC28及びCMOS30を集約したものである。82360SLでは、システム割り込みはSMI（システム・マネジメント・インタラプト）と呼ばれている。SMIが発生すると、メモリ・コントローラはCPU10とPMメモリ13とのコミュニケーションを可能にし、SMIハンドラであるところのPMCの実行が開始される。SMIハンドラ（PMC）は、SMIの原因を突き止め、原因に応じた処理ルーチンにジャンプする。

【0017】シリアル・ポート24は、シリアル・ポート・バッファ34を介して1以上のI/Oデバイスと接続される。それらのI/Oデバイスについては、ポート24に割り付けるI/O空間のベース・アドレスを、任意に（例えば3F8（H）または2F8（H））に設定することができる。

【0018】RTC28とCMOS30は1つのチップに搭載されている。そのチップには、システムのパワーがオフのときでも、リザーブ・バッテリー36からパワーが供給される。リザーブ・バッテリー36はコイン・バッテリーであってよい。

【0019】ROM32には、BIOSコードの他に、PMCがストアされている。システムのPOR時に走るPOST（パワーオン・セルフ・テスト）によって、PMCはROM32からPMメモリ13にロードされる。

【0020】CPU10は、KMC（キーボード／マウス・コントローラ）38を介して、マウス42及びキーボード44からの信号を受信する。本実施例では、キーボード44のマトリックスをモニタするプロセッサ（サブCPU40）が、パワーマネジメント機能の一部も担当している。サブCPU40は、キーボード44のマトリックス、リッド46、メインバッテリー48をモニタし、それらデバイスに関して所定の状態が発生したこと、例えばホット・キーが押されたり、リッドが閉じられたり、メイン・バッテリー48がロー・バッテリー状態になったことを検出すると、信号線50をアクティブにする。サブCPU40はバス41を介してもバス14と結ばれており、バス41を通じてメインCPU10との間でパワー・マネジメント（ハイパーネーションを含む）に関する命令及びデータをやりとりする。

【0021】サブCPU40は、デバイスごとにパワーオフしたり、あるいはシステム全体をパワーオフすることを指示するために、パワー・コントロール・レジスタ54に信号を出力するが、その詳細は後で図2を参照して説明する。

【0022】CPU10は、バス14を介してVGAチップ56と結ばれている。VGAチップ56はディスプレイ

レイ・コントローラであり、VRAM58の内容にしたがって情報が表示されるように、LCDC（LCDコントローラ）60を通じてLCDパネル62を制御する。あるいは、CRT66とDAC（デジタル・アナログ・コンバータ）64を含むディスプレイ・ユニットをシステムにオプションとして装着することもできるが、その場合でも、CRT66への情報の表示はVGAチップ56が制御する。

【0023】システムには、外部記憶装置として、ハード・ディスク装置68とFDC（フロッピー・ディスク・コントローラ）70/FDD（フロッピー・ディスク・ドライブ）72とが搭載されている。ハード・ディスク装置68は、ハード・ディスク・ドライブ及びそれに搭載されたハード・ディスクを含む概念であり、以下ではハード・ファイルとも呼ぶ。ハイパーネーション・モードに入るときにデータをストアするファイル（ハイパーネーション・ファイル）はハード・ファイルに作られる。本発明によれば、ハード・ファイルが取り外し可能（リムーバブル）であっても、ハイパーネーション/ウェーク・アップをサポートすることが可能である。

【0024】以上のハードウェア要素の他に、実際には多くの1/F（インターフェース）が存在する（例えば、ハード・ファイル68とバス14の間には、バス・トランシーバが存在する）けれども、それらは当業者には周知の事項であるので、説明を簡潔にする目的から、図示を省略している。

【0025】次に、図2を参照して、パワー・オン/オフ機構を説明する。メイン・バッテリーの出力はDC/DCコンバータ74を経て、システム全体を一度にパワーオフすることのできるFETスイッチ76に入力される。FET76の出力は、メイン・メモリとVRAMに対しては直接供給される。他方、LCDバックライト用パワー入力端子に対しては、FETスイッチ80を介して、シリアル・ポート24に直結されている内蔵モデムに対してはFETスイッチ81を介して、メインCPU10及びその他の周辺装置に対してはFETスイッチ78を介して、それぞれFET76の出力が供給される。

【0026】それらFETスイッチの各々は、パワー・コントロール・レジスタ54の対応するビット・セルと電氣的に接続されている。したがって、サブCPU40がレジスタ54にセットする値によって、FET76、78、80、81のオン/オフが制御される。ハイパーネーション・モードに入るときは、PMCの指示を受けてサブCPU40がFET76をオフにする値を対応するビットにセットし、メイン・メモリとVRAMを含むシステム全体のパワーをオフにする。サスペンド・モードに入るときは、PMCの指示を受けて、サブCPU40が、FET76をオンにし、FET78、80、81をオフにする値をレジスタ54にセットし、メイン・メモリとVRAMを除くシステムのパワーをオフにする。

【0027】パワー・コントロール・レジスタ54のクリア端子とシステムのパワースイッチ82とは電氣的に接続されている。したがって、ユーザがシステムのパワースイッチをオンにしたときに発生する信号によって、レジスタ54の値はリセットされ、すべてのFETスイッチがオンになり、システム全体にパワーが供給される。

【0028】B、ハイパーネーション・ファイルの構造図3に示すように、実施例では、ハイパーネーションを実現するために、制御情報のブロックA、ファイル・アロケーション情報のブロックB、PMメモリ中の作業データのブロックC、VRAMの内容のブロックD、及びメイン・メモリの内容のブロックEのための場所を、ハード・ディスク上に確保する。制御情報は、後で詳述するように、システム・コンフィギュレーション情報やブロックB～Dの各々のスタート・アドレスなど、パワーオン直後に必要となる情報である。作業データは、ハイパーネーションに必要なその他のデータであり、例えば、ハードウェア・コンテキスト情報（後述する）や種々の制御フラグである。制御フラグの1例は、ハイパーネーション・モードに入るとき及びそこから出るときにブザーを鳴らすかどうかを選択するための、ユーザによってその値が変更可能なフラグである。

【0029】ブロックA～Eは、物理的に一続きの領域であってもよい。しかしながら、少なくともブロックAは、ディスク上の固定された場所に存在する必要がある。そこで、実施例では、ブロックAのみを、ハード・ディスクの最も内側に定義されるCEシリンダ（リザーブされた、ユーザがアクセスできないシリンダ）の先頭に配置する。要は場所が固定されていればよいのであって、制御情報をCEシリンダの途中のセクタからストアすることも可能である。

【0030】セクションCで詳しく述べるように、ブロックB～Eは、OSのファイル・システムを用いて、ハード・ディスクのユーザ・パーティションに、ユーザ・ファイルと同じ次元で、1つのファイルとして、その場所が確保される。そのファイルの名前はリザーブされており、本実施例ではPM_HIBER. BINである。ブロックB～Eはそれぞれ可変長であり、ハイパーネーション・ファイルに実際にデータがストアされた時点で、ブロックC～Eのスタート・アドレスが決まる。通常、ブロックB～Eを構成するセクタは物理的にハード・ディスク上でとびとびに存在する。セクタの接続情報は、ハード・ディスク上のファイル・アロケーション情報領域（OSがDOSの場合はFAT）にリストの形で記録される。セクションDで詳しく述べるが、PMCは、PM_HIBER. BINを構成するセクタの接続情報を独自のアロケーション情報に変換して、ブロックBにストアする。

【0031】C、ハイパーネーション・ファイルの生成

図4乃至図6を参照して、本発明によるハイパーネーション・ファイルの作成を説明する。

【0032】本発明では、OSファイル・システムを利用して、ハード・ファイルなどのブロック・デバイスのユーザ・パーティションにハイパーネーション・ファイルを設ける。そのために、ファイル作成ユーティリティ（例えば実行可能プログラム（.EXEファイル）である）を用意して、ハイパーネーション・ファイルを作成する。しかしながら、ブロック・デバイスが複数存在するとき、必ずしもPMCがそのすべてのデバイスに直接（OS管理下のドライバ／BIOSを経由しないで）アクセス可能とは限らない。なぜなら、ユーティリティからは論理的に同じドライブに見えても、I/F（ハードウェア）が適当でない等の理由で、PMCが直接アクセスできない場合があるからである。

【0033】図4の例だと、OSファイル・システムは、リムーバブル・ディスク（光ディスク、SSF（ソリッド・ステート・ファイル）、SRAMカードなど）、ネットワーク・ドライブ（リモート・ファイル）、RAMディスク、圧縮パーティション、SCSIドライブであるハード・ファイル1、及びIDEドライブであるハード・ファイル2のパーティション1、2、3に対してアクセス可能であり、それらはユーティリティにとって同等の論理ドライブである。

【0034】それらのドライブについて説明すると、まず、ネットワーク・ドライブは、ネットワークを介して接続された異なるシステムのドライブであり、PMCがハイパーネーション／ウェーク・アップのためのデータ転送を行うためにアクセスすることが不可能である。RAMディスクは仮想的なドライブであり、PMCがアクセスできるようなI/Fは用意されない。（そもそもRAMディスクは揮発性であるから、ハイパーネーション・モードの間、データを保持することができない。）

【0035】圧縮パーティションは、圧縮されたデータをストアするドライブであり、特別なアルゴリズムを持った対応するドライバが、当該ドライブに書き込まれるデータを圧縮し、あるいは読み出されるデータを伸長するものである。PMCは、その圧縮／伸長アルゴリズムを利用することができない。なぜなら、その圧縮パーティションに対応するドライバは、OS管理下でのみ動作するからである。したがって、圧縮パーティションはハイパーネーション・ファイルの作成に不適當である。また、PMCが直接アクセスできるハード・ファイル用I/FとしてIDEドライブしか用意されていないければ、PMCはハード・ファイル1にアクセスできない。

【0036】ハード・ファイル2に作られたパーティション4は隠しパーティションである。また、パーティション5は、OSにサポートされていないフォーマットのドライブである。例えば、OSがDOSであるとき、OS/2のHPFSフォーマットはサポートされない。

（OS/2は、インターナショナル・ビジネス・マシーンズ・コーポレーションの商標である。）これらパーティション4、5は、PMCにとってアクセス可能であっても、OSファイル・システムにとってはアクセス不可能である。

【0037】それらのドライブの中には、ユーティリティがOSファイル・システムに対して問い合わせることによって、ハイパーネーション・ファイル作成に不適當であることが判るものがある。図4の例では、ネットワーク・ドライブがそうである。しかしながら、ユーティリティにとって、その他のドライブに対するPMCのアクセス可能性は不明である。

【0038】ドライブに割り当てられるドライブ・レターでもってユーティリティがハイパーネーション・ファイル作成ドライブを特定することが考えられる。しかしながら、ドライブ・レターは、ドライブをサポートするドライバがインストールされる順序によって異なる。また、ブート可能なドライブが複数ある場合には、割り当てられるドライブ・レターは、ブートするドライブがどれかによって異なってくる。したがって、ユーティリティがドライブ・レターを使って固定的にドライブを特定することができない。一方、PMCもまた、ドライブとドライブ・レターの対応関係を知ることができない。

【0039】本発明は、PMCがアクセスでき、ユーティリティがハイパーネーション・ファイルを作るのに適したドライブを、次のようにして判別する。

【0040】まず、図5を参照して、POR時に行われるシステムの動作を説明する。パワースイッチがオンされたり、あるいはシステムがリセットされると、POSTプログラムが走る（ステップ502）。POSTがPMCをPMメモリにロードし終わると、一時的にPMCが実行され、PMCがアクセス可能なドライブの各々で、ハイパーネーション・ファイル（PM_HIBER、BIN）を探す（ステップ503）。探索が終了すると、再びPOSTが実行され、残りの処理を行った後、ノーマル・ブートあるいはウェーク・アップのためのブートの手順に入る（ステップ504、505）。

【0041】次に、図6を参照して、ハイパーネーション・ファイル作成ユーティリティが実行するステップを述べる。ユーザがコマンドをキー・インしたり、グラフィカル・ユーザ・インターフェースを介して指示を与えることによって、ハイパーネーションがイネーブルされると、ファイル作成ユーティリティの実行が開始される（ステップ601）。ステップ602で、ユーティリティは、BIOSをコールして、ハイパーネーション・ファイルに必要とされるサイズ（メイン・メモリ、PMメモリの作業データ領域、及びVRAMのサイズの合計）を知る。

【0042】ステップ603では、PMCに対し、上記必要サイズ以上のサイズを持ったハイパーネーション・

ファイル (PM_HIBER. BIN) がステップ 503 での探索により見つかったかを尋ねる。そのようなファイルが存在すれば、データをストアするのにそれを使えばよいので、以後のステップを打ち切る。

【0043】PMCの返答が否定的であった場合は、ネットワーク・ドライブのように不適当であることがOSファイル・システムに問い合わせることによりわかるドライブを除き、ユーティリティがアクセス可能なドライブの各々について、以下のステップを実行する。

【0044】まず、選んだドライブのサイズが、上記必要サイズ以上か否かを判断する (ステップ 606)。判断結果が肯定的であるならば、OSファイル・システムを使って、そのドライブにリザーブされた名前を持つ小さなファイルを一時的に作成し、一時ファイルが作られたことをPMCに知らせる (ステップ 607)。その一時ファイルの名前はPM_HIBER. BINであってもよいし、その他の名前であってもよい。作るファイルのサイズは0であってもよい。

【0045】ユーティリティから知らせを受けたPMCは直ちにその一時ファイルを読むことを試み、成功した場合にはユーティリティにコンファメーションを送る。PMCからコンファメーションを受け取った場合、そのドライブはPMCがアクセスでき、かつサイズ的にも十分である。したがって、ユーティリティは、まずその一時ファイルを削除し、しかる後、上記必要サイズと同じサイズを持ち、名前がPM_HIBER. BINであるハイパーネーション・ファイルを、OSファイル・システムを用いてそのドライブに作成する (ステップ 609)。

【0046】PMCからコンファメーションが来なかった場合には、一時ファイルを削除し (ステップ 611)、次に選択したドライブについてステップ 606、607、608を繰り返す。ドライブのサイズが不十分である場合には、ステップ 607、608、611をスキップする。どのドライブについてもPMCからコンファメーションを受け取ることができない場合には、ハイパーネーションが実行不可能であることを、ブザーを鳴らしたり、あるいはメッセージを表示するなどの手段により、ユーザに知らせ、一連の動作を終える (ステップ 614)。

【0047】以上の例では、ハイパーネーション・ファイルを作成するのに適当なドライブが見つかる、直ちにそのドライブにハイパーネーション・ファイルを作成した。これの変形例として、すべての論理ドライブについてハイパーネーション・ファイル作成の適否をチェックした後、ユーザに対して適当であるドライブを提示するようにしてもよい。その場合には、ユーザに対して、提示されたドライブを選択するようにプロンプトし、ユーザが選択したドライブにハイパーネーション・ファイルを作成する。

【0048】

D. ストア／リストア・シーケンスの概要

図7を参照しつつ、ハイパーネーション・モードに入るとき及びそこから出るときのストア／リストア・シーケンスの概要を説明する。図示したステップ701から708までの処理、及びステップ713から719までの処理は、PMメモリのPMCをメインCPUが実行することによって行われる。

【0049】まず、ストア・シーケンスについて述べる。図1に示したように、サブCPU40によって所定の状態の発生 (例えば、ホット・キーが押されたり、ロー・バッテリー状態になったこと) が検出されると、トラップ・ロジック16からメインCPU10に対してシステム割込み信号が送られる。これをトリガーとして、現在実行中のタスクは中断され、システムの動作の支配権はOSまたはアプリケーションからPMC (トラップ・ハンドラ) に移る。

【0050】PMCは、システム割込みの原因を分析し、信号線50を通じて知らされる外部事象が原因であると判断したときには、ハイパーネーション・モードに入るべく、ストア動作実行ルーチンにジャンプする。

【0051】まず、PMCは、I/Oデバイスのアクティビティの有無をチェックする (ステップ701)。アクティビティが存在するとき (例えばDMAが行われているとき) には、所定時間 (例えば10msec) 経過後、再度アクティビティをチェックすることを繰り返し、I/Oアクティビティが検出されなくなるまで待つ。

【0052】I/Oアクティビティが検出されないとき、PMCは、まずハードウェア・コンテキスト情報をPMメモリにセーブし、次にハードウェア・コンテキスト情報を含む作業データ全体をPMメモリからハード・ディスク装置にセーブする (ステップ702)。ハードウェア・コンテキスト情報の代表例は、CPU、割込みコントローラ、DMAコントローラ、ビデオ・コントローラ等の各チップのレジスタの値や、タイマーのカウント値である。

【0053】ステップ703で、PMCは、VRAMのオリジナル・データをハード・ディスク装置に転送する。このとき、VRAMのデータを圧縮してからハード・ディスクにストアしてもよい。ステップ704では、ハイパーネーションのためのセーブ動作中であることを示すアイコンのイメージをVRAMに書き込んで、ディスプレイ装置に表示する。

【0054】アイコンを表示している間に、PMCは、メイン・メモリのオリジナル・データをハード・ディスク装置に転送する (ステップ705)。このとき、メイン・メモリのデータを圧縮してからハード・ディスクにストアしてもよい。ステップ706では、システム・コンフィギュレーション情報をCMOSからハード・ディ

スク装置にセーブする。システム・コンフィギュレーション情報の代表例は、システム（ポータブル・コンピュータ）本体に接続されているオプション・デバイスのタイプ及び個数並びにメイン・メモリのサイズである。

【0055】ステップ707で、PMCは、以上のシークエンスを経てきたことを示すハイパーネーション・シグニチャをセットする。ハイパーネーション・シグニチャは、2バイトの情報であり、図の領域Aにストアされる制御情報の1アイテムである。

【0056】最後に、PMCはサブCPUに対してコマンドを送り、システムのパワーをオフにする（ステップ708）。

【0057】次に、リストア・シークエンスを説明する。システムのパワーがオンになると、メインCPUは、ROMにストアされているPOST（パワー・オン・セルフ・テスト）プログラムを実行する（ステップ712）。POSTは次のことを行う。

【0058】（i）パワーオフの間にメモリが増設されたり、あるいはFDDの数が増えたりして、システム・コンフィギュレーションが変更されていた場合には、POSTがこのことを検出して、直接的に、あるいはセット・アップ・プログラムを使って間接的に、CMOSのシステム・コンフィギュレーション情報を書き換える。

【0059】（i i）FDDに関連する動作として、FDD/FDCが正常に動作するか、ハイパーネーション・ウェーク・アップよりも優先してFDDからブートする必要があるのか、FDDが取り外されて他のデバイスやユニットが装着されていないか、などのチェックを行う。

【0060】（i i i）PMCをROMからPMメモリに転送する。

【0061】（i v）ハイパーネーション・シグニチャのチェックを行う。

【0062】FDDからブートする必要がある場合、及びステップ701～708を経てパワーオフされたのではない場合には、システム動作の支配権がPMCに移ることなしに、ノーマル・ブートのための手順に入る。FDD/FDCが正常に動作しないなど、ハードウェア・エラーを検出したときには、システムがストップする。

【0063】このように、本発明では、ハイパーネーション・シークエンスを経たか否かという履歴情報をハード・ディスク装置だけに残し、システムの他の要素には残さない。POR時に、POSTは、ハード・ディスク装置の履歴情報に基づいて、ハイパーネーション・ウェーク・アップかノーマル・ブートかを判断する。したがって、取り外し可能ハード・ディスク装置を持ち運び、ストア動作が行われたマシンと同様の機能を備えた他のマシンでタスクを再開することが可能になる。即ち、凍結されたシステム環境を自由に移動させることができ

る。

【0064】ハイパーネーション・シグニチャ、したがってハイパーネーション・コンテキストの存在が確認されると、システムの支配権はPOSTからPMCに移る。まず、PMCは、ハード・ディスクとCMOSのシステム・コンフィギュレーション情報を比較する（ステップ713）。不一致が検出されたときには、アイコン等によりエラー・メッセージを表示し、ユーザに対して、現在のハイパーネーション・シグニチャを無効にすること、あるいはパワーをオフにして変更前のシステム・コンフィギュレーションを回復することの何れかを選択するようにプロンプトする（ステップ714）。なお、システム・コンフィギュレーションの回復をプロンプトするときには、変更前のシステム・コンフィギュレーションを提示して、ユーザをガイドしてもよい。

【0065】ハード・ディスク装置が取り外し可能であるときには、ハード・ディスクにデータをセーブしたときの環境（第1の環境）とウェーク・アップのときの環境（第2の環境）が異なる可能性が高い。例えば、第2の環境の方がメイン・メモリのサイズが小さいことがある。また、第1の環境ではI/Oデバイスのベース・アドレスが特定の値であることを要求するアプリケーションが走っていたのに、第2の環境ではその値になっていないことがある。また、第1の環境ではフロッピー・ディスクにアクセスするアプリケーションを実行していたのに、第2の環境ではフロッピー・ディスク・ドライブが1台もないこともある。以上のような場合には、ウェーク・アップ自体が不可能であったり、再開されたタスクによってデータが破壊されるなどの不都合が生じる。したがって、システム・コンフィギュレーションをチェックする機能は重要である。

【0066】ウェーク・アップできる環境にあることが確認されたとき、PMCは、リストア（ウェーク・アップ）中であることを示すアイコンのイメージをVRAMに書き込んで、ディスプレイ装置に表示する（ステップ715）。ハイパーネーション・アイコンを表示している間に、PMCは、メイン・メモリのオリジナル・データをハード・ディスクからリストアする（ステップ716）。その後、アイコンを消して、VRAMのオリジナル・データをハード・ディスクからリストアする（ステップ717）。ステップ718では、ハードウェア・コンテキスト情報を含む作業データをPMメモリにリストアし、ハードウェア・コンテキスト情報はそこからさらにI/OデバイスやCPUのレジスタなどの宛先へリストアする。最後に、PMCは、FDDチェンジ・ライン・エミュレーションのためのセッティングを行う（ステップ719）。以上のシークエンスの後、システムの支配権はOSあるいはアプリケーションに移り、中断時点からタスクの実行を再開する。

【0067】E. ハイパーネーション・ファイルへのセ

ープノリストア動作の詳細

図8を参照して、ハイパーネーション・ファイル (PM__HIBER. BIN) へのデータのセーブ動作 (図7のステップ702、703、705、706、707) をより具体的に説明する。

【0068】まず、PMCは、ハードウェア・コンテキスト情報をメモリのPMC領域にストアする (ステップ801)。このように、最初にハードウェア・コンテキスト情報をセーブしておけば、後のセーブ動作をやりやすくするためにハードウェア・コンテキストを変更することが可能になる。

【0069】ステップ802では、ハード・ディスクのマスター・ブート・レコード (MBR) にアクセスし、ハード・ディスク装置のパーティション情報 (各パーティションのスタート・アドレスとサイズ) を獲得する。MBRは、ハード・ディスクの最も外側に定義されるシリンドラであり、リザーブされた、ユーザがアクセスできないシリンドラである。

【0070】ステップ803で、PMCは、サイズが所定値を下回るパーティションを除いた各パーティションのルート・ディレクトリのアドレスを獲得する。アドレス算出方法は公知であるので、詳しくは述べないが、例えばOSがDOSであるときは、パーティションの先頭近くにあるバイオス・パラメータ・ブロック (BPB) を参照して、そのパーティションのルート・ディレクトリのアドレスを算出する。

【0071】ステップ804では、アドレスを算出したディレクトリに順次アクセスし、リザーブされているファイルの名前 (PM__HIBER. BIN) を探す。もし何れのディレクトリにおいてもPM__HIBER. BINが見つからなかったときには、ハイパーネーションの実行を拒否し、そのことをブザーを鳴らしたりメッセージを提示するなどの手段でユーザに知らせる。

【0072】PM__HIBER. BINが見つかったときは、そのファイル・アロケーション・リストを追跡する (ステップ805)。図9は、OSがDOSであって、ハード・ディスクのフォーマットが4セクタを1クラスタとするときのファイル・アロケーション・リストを、PM__HIBER. BINがクラスタ100から始まりクラスタ149まで続くセクタ・ブロックと、クラスタ500から始まりクラスタ549まで続くセクタ・ブロックと、クラスタ300から始まりクラスタ399まで続くセクタ・ブロックとに分かれている場合を例にとって示したものである。ルート・ディレクトリのPM__HIBER. BINのエントリには、最初のクラスタ番号100が記述されている。周知のように、クラスタに1対1に対応してFAT (ファイル・アロケーション・テーブル) が設けられており、FATには後続のクラスタ番号 (したがってFAT番号) またはファイルの最後を示す特別な番号が記述されている。ステップ805

で、PMCは、ハード・ディスクに1回以上アクセスして、200個のFATのリストを追跡することになる。

【0073】このようにして獲得したファイル・アロケーション情報を、ステップ806では、ストアノリストアに適した独自のアロケーション情報に変換する。図10は、その変換後の情報のフォーマットを示す。図示した例は図9に対応しており、PMCは、一続きのセクタ・ブロックごとにその先頭のセクタ・アドレス (ディスク上の物理的な先頭セクタ・アドレスからの相対アドレス) と長さ (セクタ数) を記録した8バイトのデータを生成する。変換後のアロケーション情報は一旦PMメモリの作業データ領域にバッファされる。

【0074】再び図8を参照するに、ステップ807で、PMCは、PM__HIBER. BINのファイル・サイズが、現在搭載されているVRAM、メイン・メモリ及びPMメモリの作業データ領域のサイズの合計をサポートできるか否かをチェックする。例えば、ハイパーネーション・ファイルを生成した後で、メイン・メモリを増設した場合には、すべてのデータをセーブすることができない。そこで、ファイル・サイズが十分でないときは、ハイパーネーションの実行を拒否し、そのことをブザーなどの手段を使ってユーザに知らせる。

【0075】ファイルが十分に大きいときは、ステップ806で生成されたアロケーション情報それ自身を使って、アロケーション情報をPMメモリからハイパーネーション・ファイルにセーブする (ステップ808)。その後、作業データ、VRAMの内容、メイン・メモリの内容を、それぞれハイパーネーション・ファイルにセーブする (ステップ809～811)。それらデータをハード・ディスクに転送する際にも、PMメモリ中のファイル・アロケーション情報が参照される。

【0076】最後に、制御情報を作成し、ハード・ディスクの制御情報領域 (図3の領域A) にセーブする (ステップ812、813)。制御情報のアイテムは、図3に示したブロックB、C、Dそれぞれのスタート・アドレス、CMOS中にある現在のシステム・コンフィギュレーション情報、及びハイパーネーション・シグニチャである。そのうち、システム・コンフィギュレーション情報には、デバイスのベースI/Oアドレス、メイン・メモリのサイズ、及びデバイス・コンフィギュレーション (デバイスのタイプと個数) が含まれる。このうち、ベースI/Oアドレスは、例えばシリアル・ポートに接続されているデバイスの各々について、そのベース・アドレスが3F8 (H) または2F8 (H) のどちらであることを示す。

【0077】次に、図11を参照して、ハイパーネーション・ファイル (PM__HIBER. BIN) からのデータのリストア動作 (図7のステップ713、716、717) をより具体的に述べる。

【0078】ハイパーネーション・ファイルのハード・

ディスク上の位置を知るために、PMCは、まず、CEシリンダの制御情報領域にアクセスし、そこにある制御情報を読む（ステップ1101）。制御情報領域のアドレスは固定されているので、直ちにアクセスすることができる。ステップ1102では、制御情報に含まれるシステム・コンフィギュレーション情報を、ウェーク・アップ・マシンのCMOS中のシステム・コンフィギュレーション情報と比較する。

【0079】ステップ1103で、PMCは、制御情報に含まれるスタート・アドレスを使って、ハード・ディスク上のファイル・アロケーション情報ブロックにアクセスし、そこにあるファイル・アロケーション情報をPMメモリにリストアする。そのファイル・アロケーション情報を用いて、PMCは、まずメイン・メモリの内容をリストアし、その後で、VRAMの内容をリストアする（ステップ1104、1105）。これらのステップでは、制御情報領域に含まれるメイン・メモリ・ブロック及びVRAMブロックのスタート・アドレスも用いられる。最後に、ハード・ディスク上の制御情報を無効にして、リストア動作が完了する（ステップ1106）。ハイパーネーション・シグニチャも無効になるので、その後で再びハイパーネーション・モードに入ることはない限り、パワーオンすれば通常の手順でブートされることになる。

【0080】なお、説明を省略したが、実際には、図8のステップ810と811の間、及び図11のステップ1103と1104の間には、ハイパーネーション・アイコンを表示するステップがあり、図11のステップ1105と1106の間には、H/Wコンテキスト情報をリストアするステップがあることに注意されたい。

【0081】ハイパーネーション・ファイルを構成するセクタのハード・ディスク上での位置情報は、他のユーザ・ファイルと同様に、OSのファイル・システムによって、複雑なリストの形で管理される。そこで、本発明は、ハイパーネーション・ファイルへのデータ転送を開始する前に、それらセクタの位置情報を獲得するべく、OSまたはドライバの管理する複雑なリストにアクセスし、それを独自のアロケーション情報に変換し、バッファ（PMメモリ）に入力する。そして、VRAMやメモリからハード・ディスク装置にデータを転送するときは、ファイルのセクタ位置を知るために専らその独自のアロケーション情報を参照する。したがって、ハード・ディスクへのデータ転送中は、ハード・ディスク上のFAT領域などのファイル・アロケーション情報領域へアクセスしなくてよく、セーブ動作を高速化することができる。

【0082】また、本発明では、セーブ動作を行うときに作成した独自のハイパーネーション・ファイルについてのアロケーション情報をハイパーネーション・ファイルの一部に書き込み、かつそのスタート・アドレスをハ

ード・ディスク上の固定されたアドレスにあるブロックに書き込んでいる。ハード・ディスク装置からVRAMやメモリにデータを転送するときには、ファイルを構成するセクタの位置を知るためにそのファイル中にストアされていたアロケーション情報を参照する。したがって、ハード・ディスクからのデータ転送中は、ハード・ディスク上のFAT領域などのファイル・スペース・アロケーション情報領域へアクセスしなくてよい。このようにして、ウェーク・アップ時において、ハイパーネーション・ファイルのアロケーション情報を獲得するためのハード・ディスクへのアクセス頻度を最小限に留めることができ、リストア動作を高速化することができる。

【0083】F. アイコン表示動作

図12を参照して、アイコン表示に係る動作（図7のステップ703～705及びステップ715～717）をより具体的に述べる。

【0084】本発明では、メイン・メモリの内容とVRAMの内容を分けて管理する。ハイパーネーション・モードに入るとき、PMCは、先に、VRAMの内容をハイパーネーション・ファイルにセーブする（ステップ1201）。VRAMのオリジナル・データを保存した後、VGAチップ（ビデオ・コントローラ）をグラフィックス・モードにセットし、アイコンのイメージをVRAMに書き込んでシステム本体に接続されたディスプレイ装置に表示する（ステップ1203、1204）。メイン・メモリからハード・ディスクへのデータ転送の間、そのアイコンは表示され続ける。

【0085】ウェーク・アップするときは、VGAチップをグラフィックス・モードにセットし、アイコンのイメージをVRAMに書き込んでディスプレイ装置に表示する（ステップ1205、1206）。アイコンは、ハード・ディスクからメイン・メモリへのデータ転送の間、表示され続ける（ステップ1207）。VRAMのオリジナル・データのリストアはその後で行う（ステップ1208）。

【0086】VRAMがアクセスされる期間（ステップ1203、1208）は、アイコンは表示されない。しかしながら、一般に、VRAMにアクセスする期間は、メイン・メモリにアクセスする期間（ステップ1204、1208）に比べて十分に短い。ステップ1203、1208でアイコンが表示されないのは一瞬であり、実際上の不都合ない。

【0087】図13は、ステップ1203で表示されるスクリーンの1例を示す。アイコン101はシステムを表しており、アイコン102はハード・ディスク装置を表している。また、アイコン103は、データ転送の向きを表している。それらアイコンはアイコン・フレーム104、105によって囲まれている。アイコン・フレーム104の内側の領域106、及び105の内側の領域107の色は、背景108と異なる色である。

【0088】図14は、図12のステップ1208で表示されるスクリーンの1例を示す。アイコン101、102の位置が入れ替わっていることを除き、図13と同じデザインである。

【0089】図15を参照して、図12のステップ1203、1206に共通するアイコン描画ステップを具体的に説明する。まず、PMCは、VRAM全体に背景色のデータをセットして、スクリーンを背景色で埋める

(ステップ1501)。次に、アイコン・フレームの内側の領域106、107を背景色とは異なる色で塗る

(ステップ1502)。図13、14に示すように、領域106、107は単純な矩形なので、プログラム(PMC)がスクリーン上でのそれらの位置を指示し、中を塗り潰すことは簡単である。ステップ1503、1504では、アイコン101、102、103のイメージ・データをPMメモリから読み、それをVRAMにセットする。アイコン・イメージはデータ量が少ないので、予めROMにストアしておき、POR時にPMメモリに書き込むようにしている。

【0090】ハイパーネーション・アイコンは静止画であってよいが、ハード・ディスクとメイン・メモリの間のデータ転送時に定期的にVRAMにアクセスし、その内容を書き直すことにより、ハイパーネーション・アイコンの表示を時間とともに変化させてもよい。例えば、矢印アイコン103を点滅させたり、セーブまたはリストアされたデータ量を示すイメージをシステム・アイコン101に付け加えることが可能である。

【0091】以上のように、本発明では、メイン・メモリ・データのブロックとVRAMデータのブロックを分けて管理する。そして、ハイパーネーション・モードに入るときにはまずVRAMのオリジナル・データをストアし、その後でメイン・メモリのオリジナル・データをストアする。また、ウェーク・アップのときは、まずメイン・メモリのオリジナル・データをリストアした後で、VRAMのオリジナル・データをリストアする。即ち、セーブ動作時とリストア動作時とでVRAMとメイン・メモリへのアクセス順序を変える。

【0092】もし、メイン・メモリ・データとVRAMデータのブロックを区別して管理しないならば、先にVRAMの内容をセーブしてからメイン・メモリをセーブする場合、ウェーク・アップ時にVRAMの内容をメイン・メモリの内容よりも先にリストアしなければならない。したがって、ハイパーネーション・モードに入るときにアイコンを表示できても、ウェーク・アップ時にはアイコンを表示することができない。逆に、先にメイン・メモリの内容をセーブしてからVRAMの内容をセーブすると、今度はハイパーネーション・モードに入るときにアイコンを表示できない。

【0093】したがって、ハイパーネーション・モードに入るときとウェーク・アップのときの両方でVRAM

のオリジナル・データを破壊することなしにアイコンを表示するためには、本出願明細書で提唱するように、メイン・メモリ・データのブロックとVRAMデータのブロックを分けて管理し、セーブ動作時とリストア動作時とでVRAMとメイン・メモリへのアクセス順序を変えることが肝要である。

【0094】

G. FDDチェンジ・ライン・エミュレーション

図16乃至図22を参照して、PMCがサポートするFDDチェンジ・ライン・エミュレーションを説明する。

【0095】図16は、図1の中から、FDDチェンジ・ライン・エミュレーションに関するハードウェア要素を抽出して示したものである。図1に示されていない要素は、FDDに挿入されているFD(フロッピー・ディスク)94、FDDとFDCの間のチェンジ・ライン92、FDC内のチェンジ・ライン・ステータス・レジスタ90及びトラップ・ロジック内のモニタすべきアドレスをストアするトラップ・レジスタ96である。

【0096】フロッピー・ディスクに関しては、通常、過去にそのフロッピー・ディスクにアクセスしたか否かによって、アクセス方法が異なってくる。既に過去にアクセス(読み/書き)があった場合、そのときに読んだファイル・アロケーション情報(OSがDOSの場合にはFAT)がOS管理下の所定のメモリ・アドレスにセーブされる。したがって、アロケーション情報をあらためてフロッピー・ディスクから読む必要がない。このため、2回目以後のフロッピー・ディスクへのアクセスが高速化される。

【0097】一般に、メイン・メモリにあるフロッピー・ディスクのアロケーション情報が有効であるか否かは、次のような機構によって判別される。FDDとFDCを結ぶチェンジ・ラインは、FDDへのフロッピー・ディスクの着脱を監視する専用の信号線であり、システムのパワーがオンされたときに自動的にアクティブになる。そして、POSTがFDCを介してFDDにアクセスし、そこにフロッピー・ディスクがあることを検出したときには自動的にインアクティブになり、フロッピー・ディスクが検出されなかったときはアクティブのままである。また、フロッピー・ディスクが挿入されるとアクティブになり、挿入されたフロッピー・ディスクへの実際のアクセスがあると、自動的にインアクティブになる。フロッピー・ディスクがイジェクトされると再びアクティブに戻る。実施例では、ハイ状態がアクティブに対応し、ロー状態がインアクティブに対応する。

【0098】チェンジ・ラインのステータスはフラグに反映される。I/Oアドレス3F7(H)を割り振られたチェンジ・ライン・ステータス・レジスタ(図16のレジスタ90)のビット7がそのフラグ(チェンジ・ライン・ステータス・フラグ)であり、値が1のときアクティブであることを示し、値が0のときインアクティブ

であることを示す。

【0099】FDCに直接アクセスするのは、BIOSまたはドライバ（例えば、OSがDOSのときはBIOS、OSがOS/2のときはドライバ）である。BIOS／ドライバは、チェンジ・ライン・ステータス・レジスタの内容を読んで、そのビット7が1のとき、メイン・メモリの中のプロッピー・ディスクのアロケーション情報を無効化（フラッシュ）し、新たにプロッピー・ディスクのファイル・アロケーション情報を読み出す。

【0100】さて、ハイパーネーション・モードに入る時点でプロッピー・ディスクがFDDに挿入されており、かつ既にそれへのアクセスも行われていた場合、そのプロッピー・ディスクのファイル・アロケーション情報がハード・ディスクにセーブされ、それがそのままウェーク・アップ時にメモリにリストアされる。

【0101】そこで、ハイパーネーション・モード中にプロッピー・ディスクの交換が行われた場合を考えると、POSTはFDDにプロッピー・ディスクがあることを検出するので、ウェーク・アップ・シークエンスが終了した時点では、チェンジ・ライン信号はインアクティブになっている。したがって、BIOS／ドライバは、リストアされた古いプロッピー・ディスクのファイル・アロケーション情報を有効であると判断して、その情報を用いて現在挿入されているプロッピー・ディスクにアクセスし、誤ったデータを読んだり、そのプロッピー・ディスクのデータを破壊するという問題があった。同様の問題は、サスペンド後のレジュームのときにも起こる可能性がある。

【0102】サスペンド中またはハイパーネーション中の媒体の交換に伴う問題は、カード型の記憶媒体（SSFなど）にも存在する。しかしながら、そのような媒体の場合には、充実したソフトウェアが用意されており、それを使って問題の解決が図られている。即ち、カードをシステムに装着したままサスペンドまたはハイパーネーションの低消費電力モードに入り、そのモードから出た場合でも、低消費電力モードに入るときにあたかもカードが抜かれたかのようにシステムに対してフェイクし、レジュームあるいはウェーク・アップするときに、あたかもカードが差し込まれたかのようにシステムに対してフェイクする。これを受けて、システムは、メイン・メモリにあるカードのファイル・アロケーション情報を無効化し、改めてアロケーション情報を獲得する。このように、カードへの給電を止める低消費電力モードに入るときとそこから出るときそれぞれで、システムに対してソフトウェアによるフェイク操作を行っている。

【0103】しかしながら、プロッピー・ディスクの場合には、チェンジ・ライン信号のステータスがハードウェア的に所定のタイミングで変化し、そのステータスをCPUが読みに行くという方式が定着している。そのような慣用されている方式との調和を図りつつ、低消費電

力モード中に行われたプロッピー・ディスクの交換に起因する上記問題を解決するためには、カード型の記憶媒体の場合とは異なる手段を講じなければならない。

【0104】そこで、本発明では、ウェーク・アップまたはレジュームのシークエンス終了後に、BIOS／ドライバが最初にチェンジ・ライン・ステータス・フラグをチェックするタイミングを捉えて、そのステータスをフェイクする。そして、プロッピー・ディスクが実際には装着されたままでも、抜かれたかのようにシステムに見せかけて、プロッピー・ディスクのファイル・アロケーション情報をフラッシュさせる。具体的には次の2つの方法がある。

【0105】（1）チェンジ・ライン・ステータス・レジスタへのアクセスをトラップして、チェンジ・ライン・ステータス・フラグの値を一時的にフェイクする。

【0106】（2）チェンジ・ライン信号をハードウェア的に操作することの可能なボードを用意する。

【0107】以下、ハイパーネーション・モードから出るウェーク・アップの場合について、上記方法の各々を説明する。ただ、本発明はサスペンド・モードから出るレジュームの場合にも適用可能であることに留意されたい。

【0108】（1）最近のポータブル・コンピュータには、I/Oアクセスをトラップする機構が備わっている。それは、図1に示した例のように、インテル社の80486SL（CPU）と82360SL（トラップ・ロジック）の組み合わせにより実現される。82360SL中のレジスタ（図16のレジスタ96）にI/Oアドレスを設定しておく、80486SLからそのI/Oアドレスにアクセスする命令が出されたときに、82360SLは、80486SLに対してシステム割込み（SMI）を発する。システム割込みに応答してハンドラ（PMC）がイネーブルになり、割込みの原因を分析し、所定のI/Oアドレスへのアクセスが原因であると判断すると、トラップ・ルーチンにジャンプする。

【0109】従来、トラップ機構は、典型的には、パワーオフしているデバイスに対してアクセスする命令が出されたときに、実際にアクセスを行う前にそのデバイスをパワーオンするために用いられていた。チェンジ・ライン・ステータスをフェイクする第1の方法は、このトラップ機構を活用する。

【0110】図17を参照して、フェイク動作に関係するステップの流れを説明する。既に詳しく述べたように、パワーオン直後に走るPOSTにより、ハイパーネーション・シグニチャが確認されると、ウェーク・アップ・シークエンスに入る（ステップ171、172）。チェンジ・ラインのステータスは、パワーオンの時点ではアクティブであるけれども、POSTによりプロッピー・ディスクがFDDに装着されたままであることが検出された時点で、インアクティブになる。

【0111】ステップ173で、PMCは、チェンジ・ライン・エミュレーションのためのセッティングを行う。具体的には、トラップ・ロジック16のレジスタ96（図16参照）に、3F7（H）の値をセットする。このステップの後、システムの支配権はOS／アプリケーションに戻る。

【0112】OS／アプリケーション実行再開後の最初のフロッピー・ディスクへのアクセスのときに、チェンジ・ライン・ステータスのフェイクが行われる（ステップ174）。図18及び図19を参照して、ステップ174をより詳しく説明する。

【0113】図19は、BIOS／ドライバのコードの一部を示す。命令MOV DX, 3F7（H）が実行されると、CPUのDXレジスタに3F7（H）がロードされる。次に、命令IN AL, DXが実行されると、I／Oアドレス3F7（H）がアクセスされ、チェンジ・ライン・ステータス・レジスタの内容がCPUのALレジスタにストアされる。このとき、システム割込みが発生し、トラップ・ハンドラ（PMC）が実行される。ハンドラは、システム割込みの原因を分析し、I／Oアドレス3F7（H）へのアクセスのトラップを処理するルーチンにジャンプする。そのルーチンは、ALレジスタのビット7の値を1にセットして、BIOS／ドライバに返す。したがって、続く命令TEST AL, 80Hによって、レジスタ3F7（H）のビット7の値は1であるとの情報がBIOS／ドライバに伝えられる。これに応答して、OS／ドライバは、メモリにあるフロッピー・ディスクのファイル・アロケーション情報をフラッシュする。そして、フロッピー・ディスク上の所望のファイルにアクセスするために、改めてフロッピー・ディスクのアロケーション情報を読み出す。

【0114】なお、BIOS／ドライバに戻る直前に、PMCは、トラップ・レジスタにセットされた値をクリアする（図17のステップ175）。したがって、2回目以降のフロッピー・ディスクへのアクセスのときには、チェンジ・ライン・ステータスはフェイクされない。

【0115】（2） 第2の方法は、図20に示されるように、ボードにハードウェア要素を追加することにより実現される。追加される要素は、レジスタ112を内蔵するI／Oポート110、I／Oアドレス・デコーダ114、ORゲート116、信号線118及び120である。レジスタ112には、特定のI／Oアドレス（1500（H）とする）が割り振られている。レジスタ112の特定のビット（ビット0）の値を示す信号は、信号線118を介して、ORゲート116の一方の入力端子に入力される。ORゲート116の他方の入力端子には、FDD92からのチェンジ・ライン信号が入力される。ORゲート116の出力端子は、チェンジ・ライン・ステータス・レジスタ90のビット7と結ばれてお

り、ORゲート116の出力がレジスタ90のビット7の値となる。I／Oアドレス・デコーダ114は、アドレス・バス14をモニタし、アドレス信号3F7（H）をデコードし、I／Oポート110に対してパルス信号を出力する。デコーダ110の出力端子は、レジスタ112のクリア端子と結ばれている。

【0116】図21を参照して、第2の方法のステップの流れを述べる。ステップ211、212は図17のステップ171、172と同じなので、説明を省略する。ステップ213では、FDDチェンジ・ライン・エミュレーションをセットするために、PMCが、I／Oアドレス1500（H）を指示して、レジスタ112のビット0の値を1にセットする。その結果、図22に示すように、チェンジ・ライン・ステータスはPOSTによるFDCへのアクセスのためにインアクティブになったままであるけれども、信号線118がアクティブになるので、信号線120のステータスはアクティブとなる。したがって、チェンジ・ライン・ステータス・レジスタ90のビット7には1がセットされる。

【0117】ステップ213の後、システムの支配権はOS／アプリケーションに戻る。OS／アプリケーションが実行されている間も、レジスタ90のビット7は値が1のままであり、チェンジ・ライン・ステータスをフェイクした状態が続く。

【0118】OS／アプリケーションの再開後のフロッピー・ディスクへの最初のアクセスのときに、BIOS／ドライバがI／Oアドレス3F7（H）を読むと、そのビット7の値は1である。したがって、OS／ドライバは、フロッピー・ディスクのファイル・アロケーション情報をフラッシュする。

【0119】アドレス3F7（H）へのアクセス・サイクルの間に、デコーダ114からレジスタ112をクリアするパルスが出力される。したがって、ORゲート116の入力はどちらもインアクティブとなり、チェンジ・ライン・レジスタ90のビット7のステータスはインアクティブとなる。このように、FDDチェンジ・ライン・エミュレーション・リセット（ステップ214）は、第1の方法と異なり、ハードウェアによって実行される。

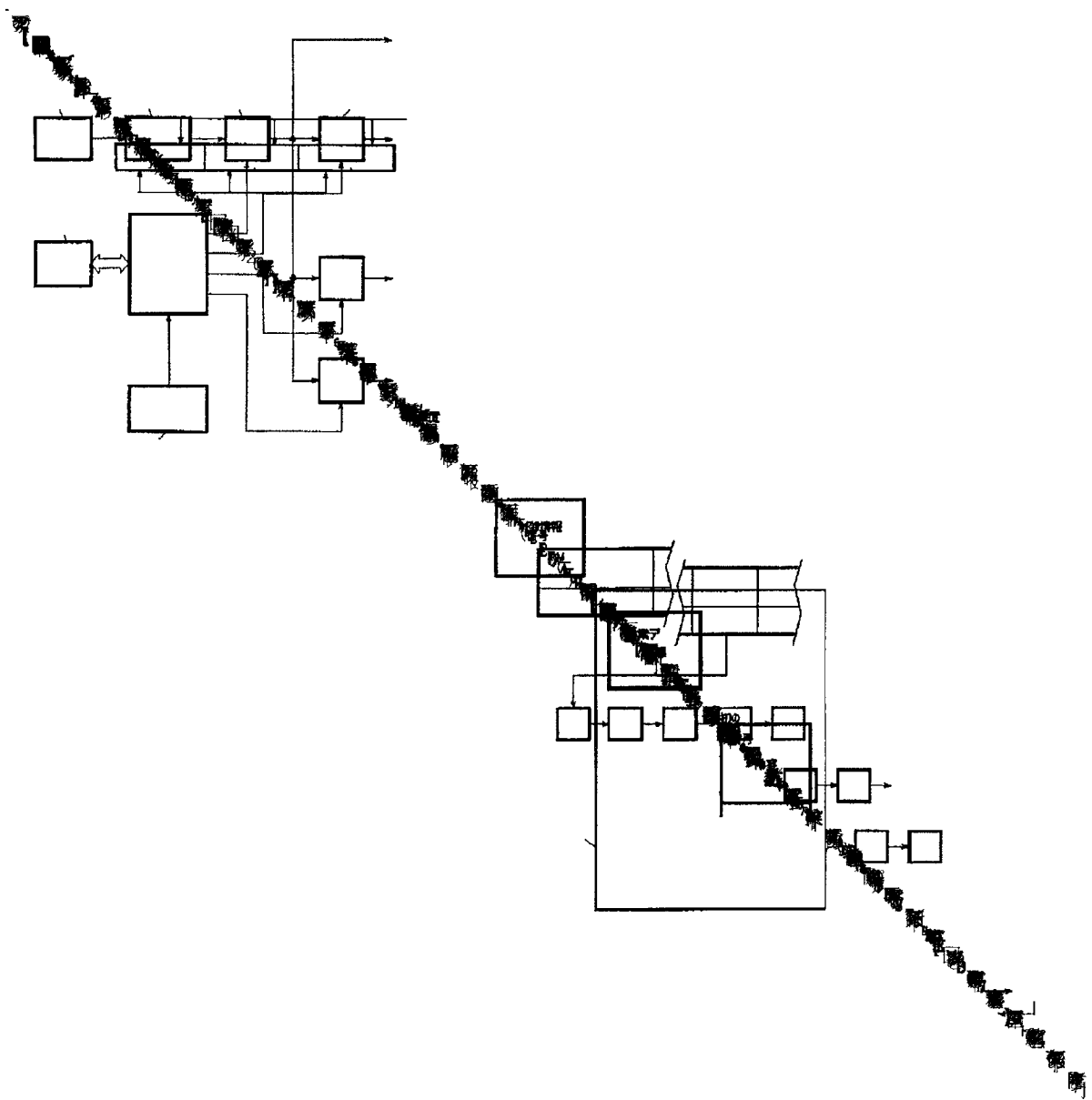
【0120】システムによっては、トラップ機構を持たないものもあり、また、トラップ機構があってもその能力が制限されているものもある。そのような場合には、第2の方法が有効である。

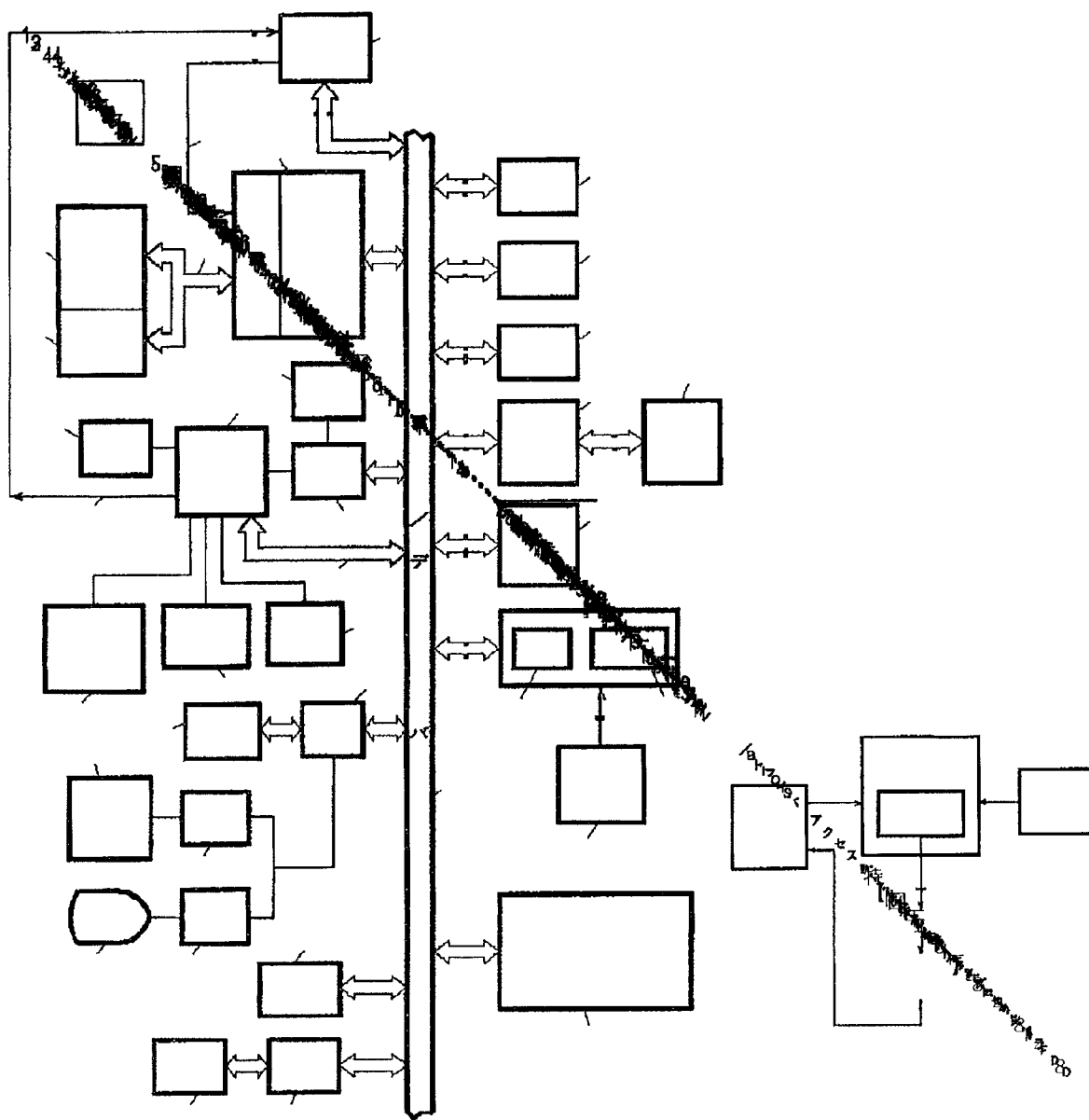
【0121】

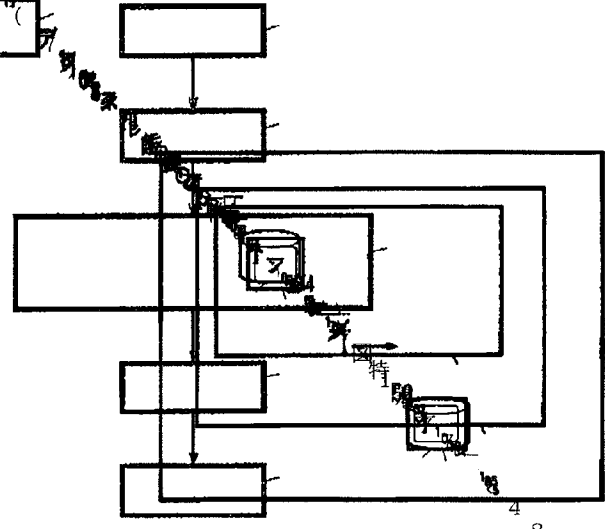
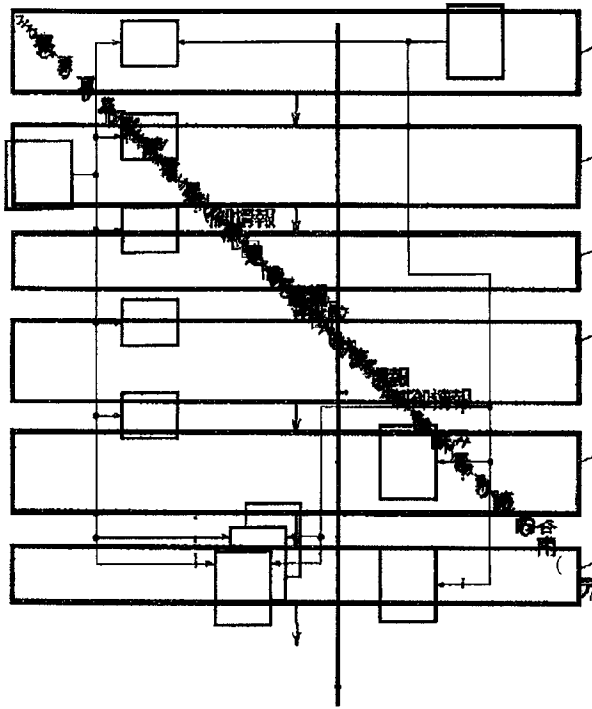
【発明の効果】本発明によれば、ハイパーネーション及びウェーク・アップのための一連の動作を高速で行うことが可能となる。

【図面の簡単な説明】

【図1】本発明を用いた情報処理システムの1例の、ハードウェア構成要素を示す図である。

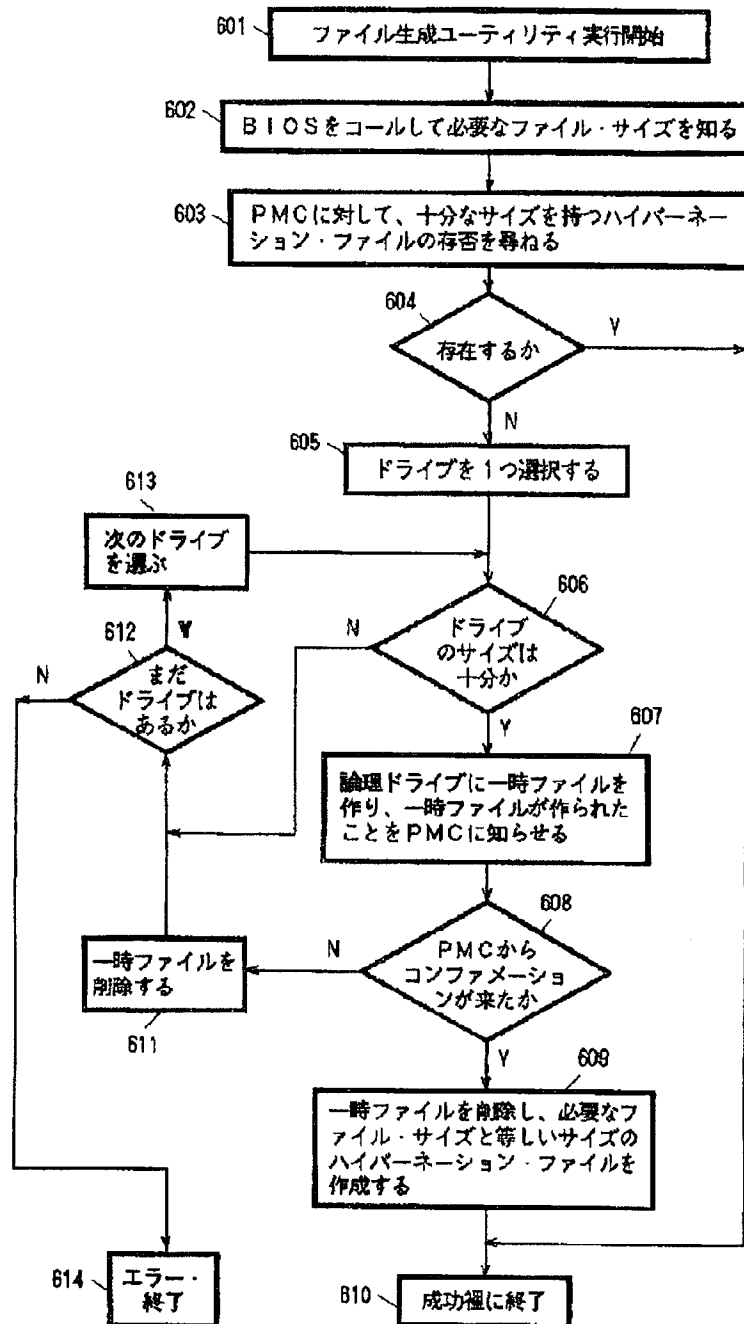




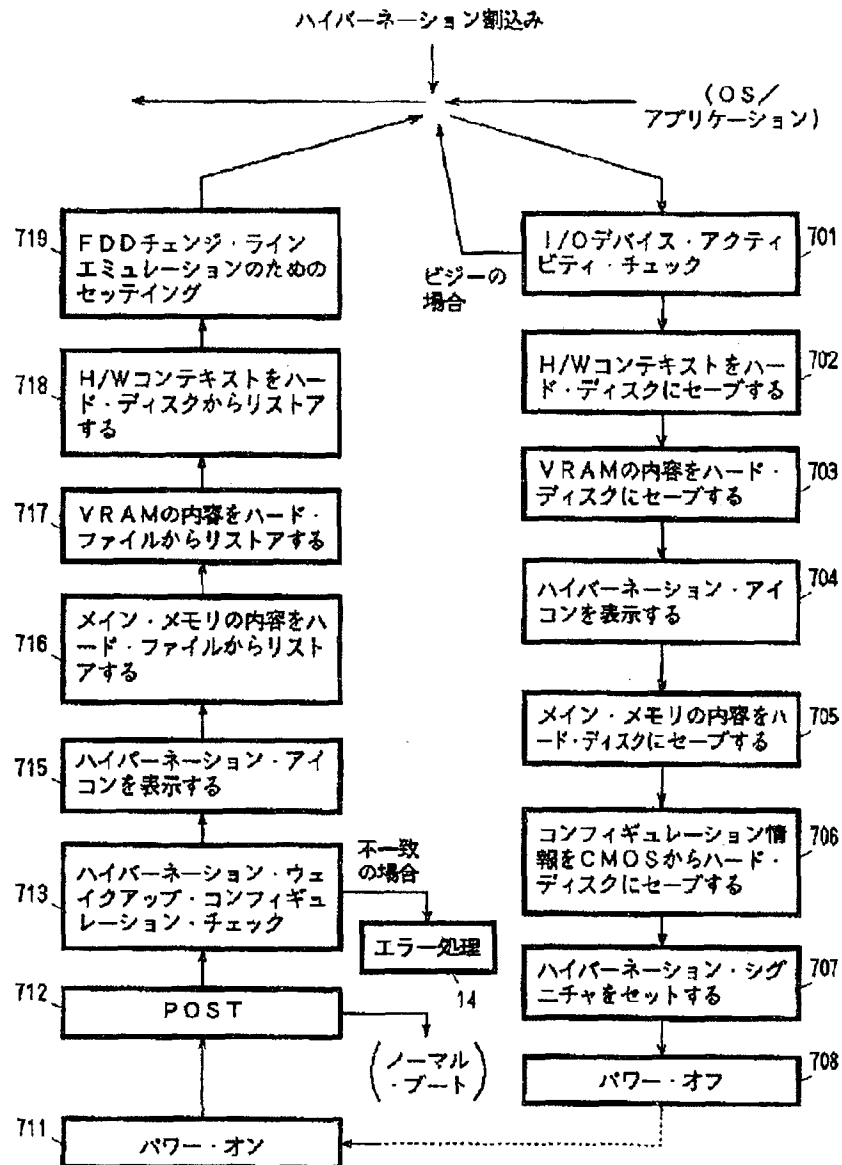


4
8
4
8.

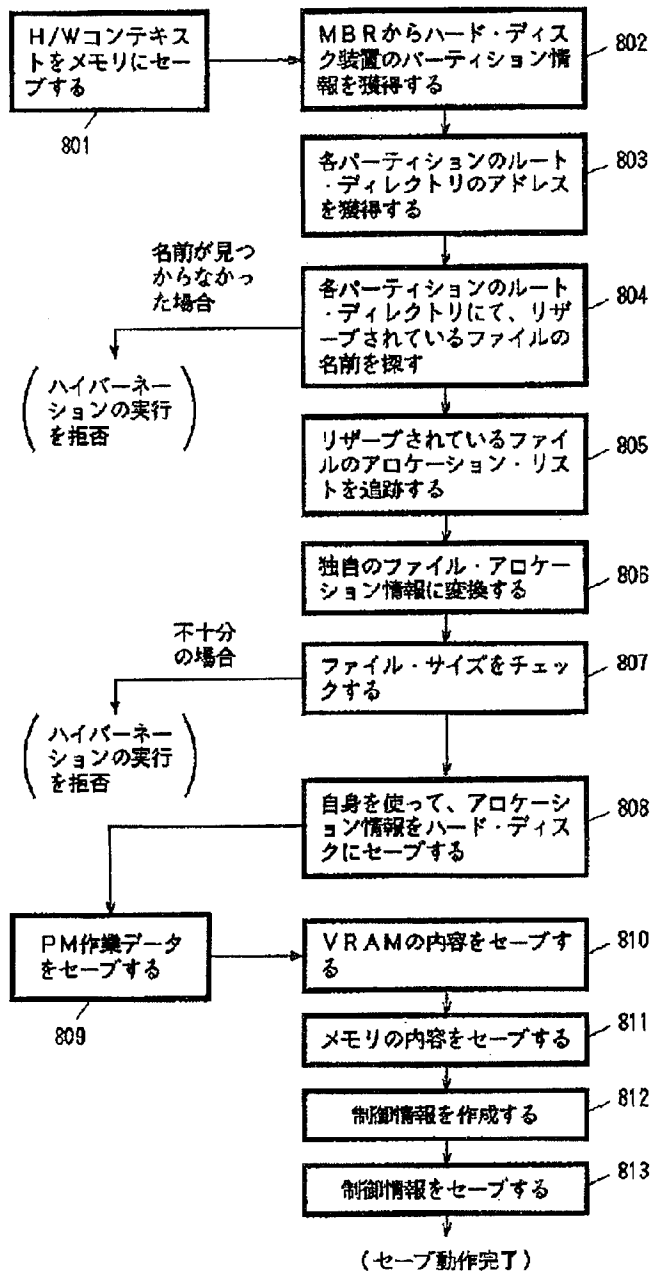
【図 6】



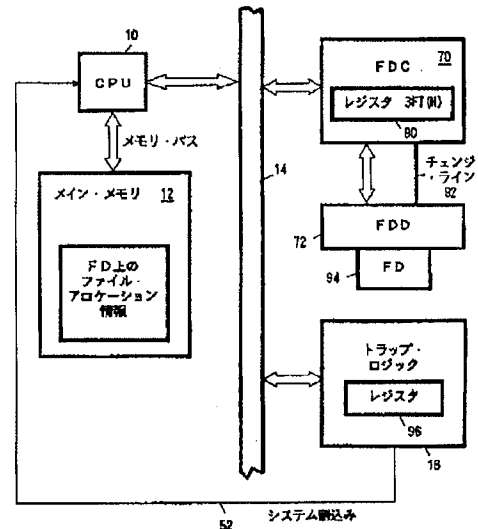
【図 7】



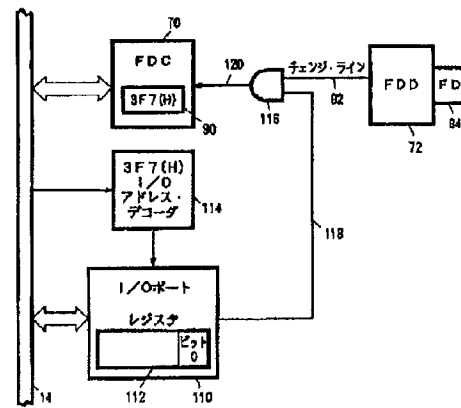
【図 8】



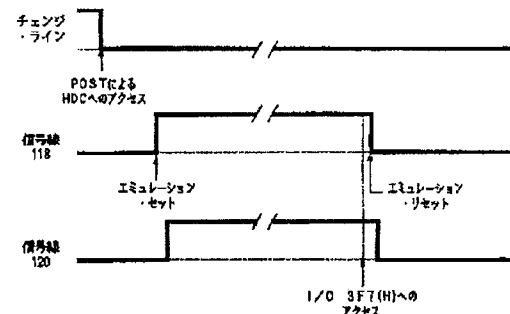
【図 16】



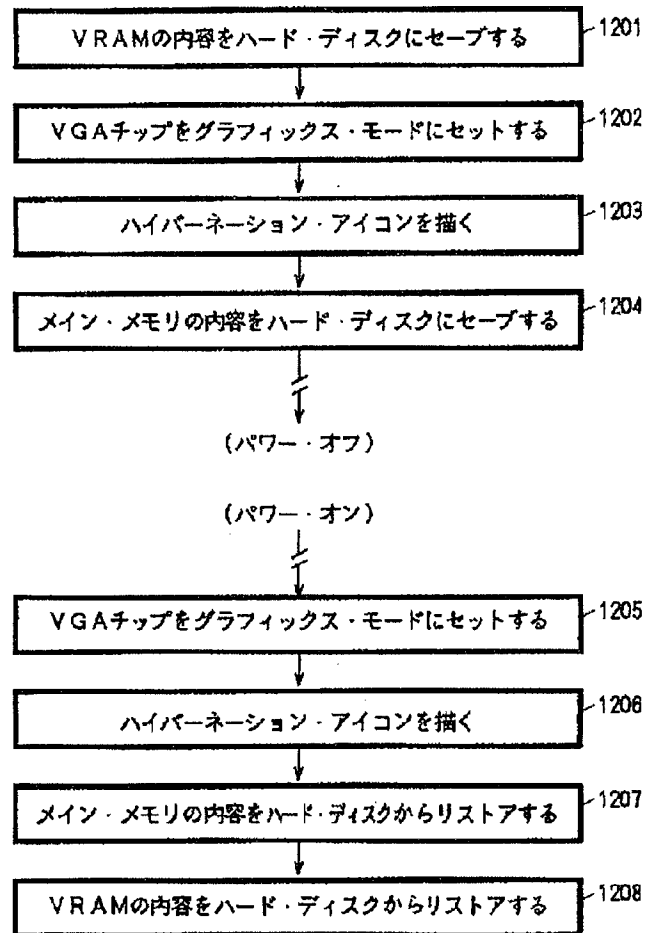
【図 20】



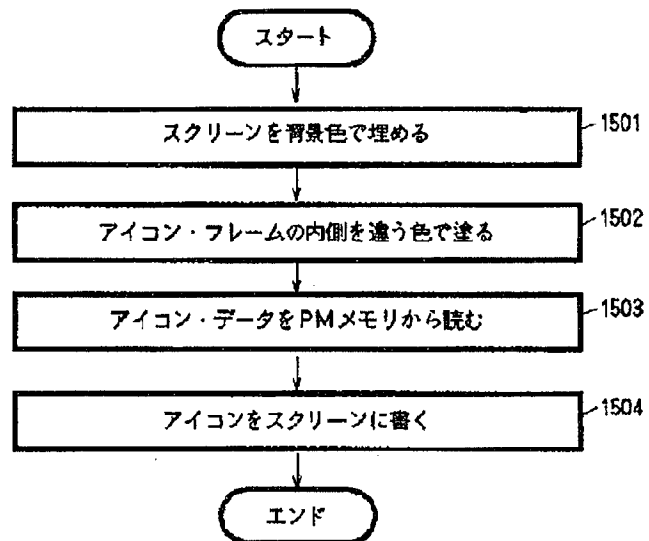
【図 22】



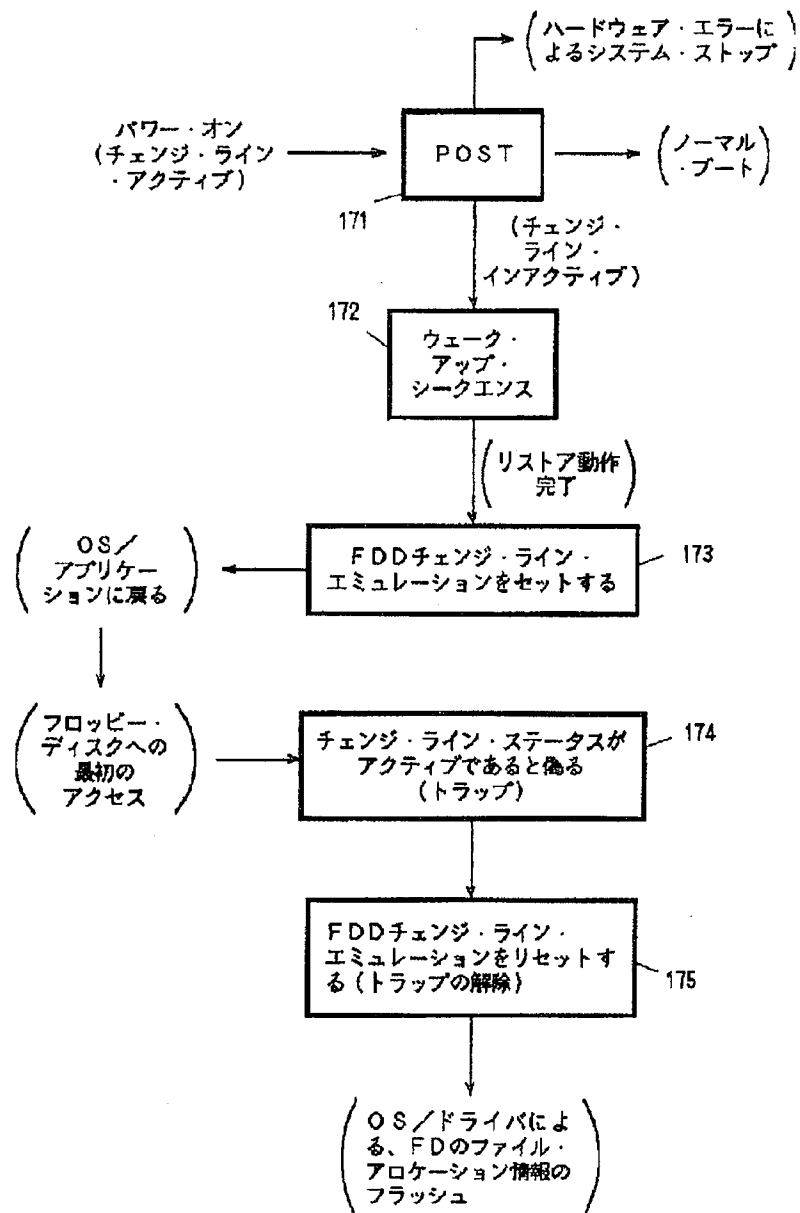
【図 1 2】



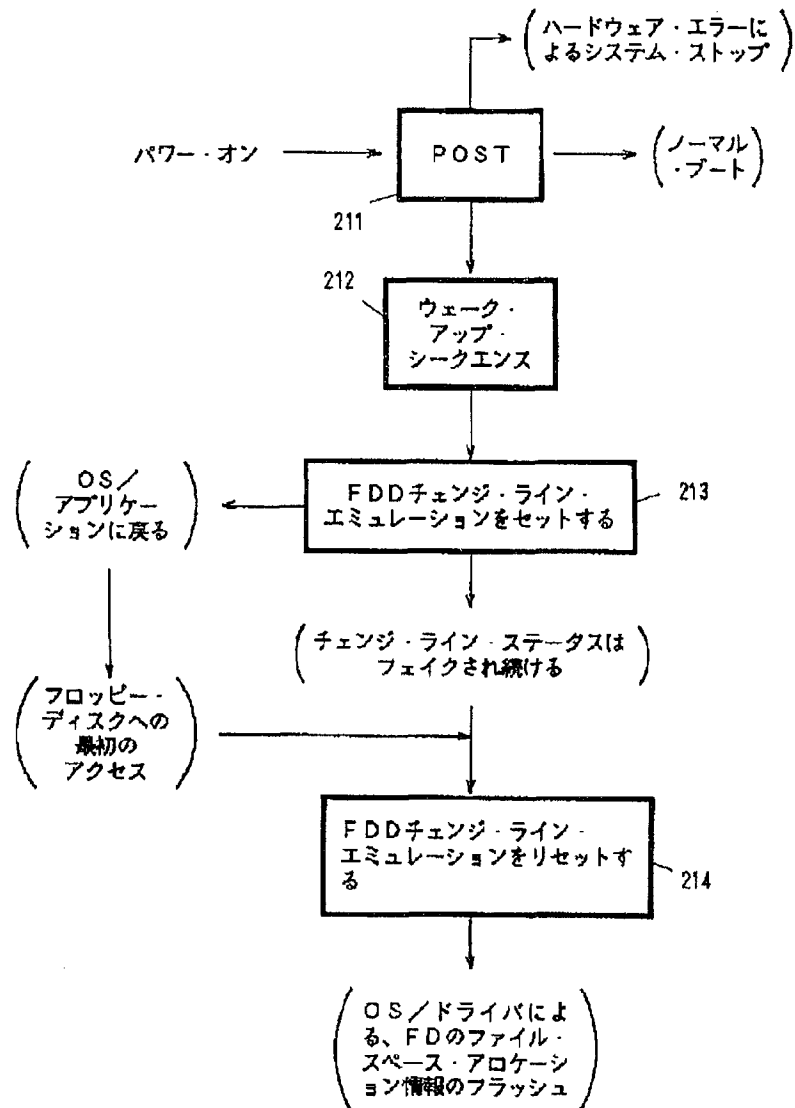
【図 1 5】



【図 17】



【図 21】



フロントページの続き

(51) Int. Cl. ⁶

G 0 6 F 12/16

識別記号 庁内整理番号

3 4 0 Q 9293-5B

F I

技術表示箇所